

**REMARKS/ARGUMENTS**

Prior to entry of this amendment, claims 1-21 were pending in this application. Claims 1, 5, 9, and 10 have been amended, no claims have been added, and claim 13 has been canceled herein. Therefore, claims 1-12 and 14-21 remain pending. Applicant respectfully requests reconsideration of these claims for at least the reasons presented below.

**35 U.S.C. § 103 Rejection, Karkare in view of Gove and Charisius**

Claims 1-21 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,266,810 to Karkare et al. (hereinafter “Karkare”) in view of U.S. Patent Pub. No. 2004/0006760 of Gove et al. (hereinafter “Gove”) and U.S. Patent Pub. No. 2002/0016954 of Charisius et al. (hereinafter “Charisius”). The Applicants respectfully submit that the Office Action does not establish a *prima facie* case of obviousness in rejecting these claims, as amended. Therefore, the Applicants request reconsideration and withdrawal of the rejection.

In order to establish a *prima facie* case of obviousness, all claimed limitations must first be taught or suggested by the prior art. *See, e.g., DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co.*, 464 F.3d 1356, 1360 (Fed. Cir. 2006). The Office Action must then provide an explicit analysis supporting the rejection. *See KSR Int’l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1741 (2007) (“a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art”). While the Office Action can use one of several exemplary rationales from the MPEP to support an obviousness rejection under *KSR*, all the rationales still require the Office Action to demonstrate that all the claim elements are shown in the prior art. *See* MPEP §2143. As will be discussed below, the references cited by the Office Action do not teach or suggest each claimed limitation. For example, none of the references, alone or in combination, teach or suggest

determining an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment as recited in independent claims 1 and 9 or a component repository configured to maintain a list of available application components that can be invoked by an integrated development environment (“IDE”) runtime environment to monitor an application, the application comprising operational logic and business logic and monitoring operation of the application component to determine a plurality of metrics associated with the application component wherein the application component is selected from the operational logic of the application as recited in independent claim 10. Furthermore, none of the references teach or suggest, alone or in combination, a policy manager of an IDE user interface environment defining a policy specifying an operational concern to be monitored and a policy agent of an IDE runtime environment enforcing such a policy to determine metrics associated with a selected application component as recited in each independent claim.

As argued previously, Karkare is directed to “the runtime profiling of a software application and the platform-dependent code component(s) used by the software application.” (Col. 1, lines 9-11) More specifically, Karkare describes “a profiling system for runtime environments that include a software application written in a platform-independent programming language, a non-application-code component invoked by the software application and a profiling tool for generating a runtime metric.” (Col. 2, lines 36-41) However, Karkare does not teach or suggest an application comprising operational logic and business logic. Rather, Karkare describes “application-code components” (see for example col. 4, line 7 – col. 5, line 35) and “non-application-code components” (see for example col. 5, line 36 – col. 6, line 42) However, Karkare does not describe the application-code components comprising business logic and operational logic. Furthermore, any reading of Karkare that would equate the application-code logic to business logic and the non-application-code logic to operational logic would not be

reasonable since, as the name “non-application-code components” implies and as explicitly stated in Karkare, “non-application-code includes any code component that does not reside within the application code component” (col. 5, lines 43-45), i.e., it is separate from the application-code component. Thus, Karkare does not describe an application comprising operational logic and business logic and further monitoring a component of such an application selected from the operational logic of the application. Therefore, the Applicants respectfully contend that Karkare does not teach or suggest determining an application component of an application to be monitored, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of an IDE runtime environment.

Also as noted previously, Gove is directed to “using an integrated development environment (IDE) and more particularly to using profiles in an integrated development environment to improve application performance.” (paragraph 2) Under Gove, a developer codes a program, the IDE executes the code and collects performance data, and the developer reviews the performance data and debugs the program. (See paragraph 20) However, Gove does not characterize or distinguish contents of the program being executed and debugged. That is, Gove does not describe the program comprising business logic and operational logic. Therefore, the Applicants respectfully contend that Gove does not teach or suggest, alone or in combination with Karkare, determining an application component of an application to be monitored, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of an IDE runtime environment.

In response to these arguments, the Office Action seems to agree that neither Karkare nor Gove teach or suggest, alone or in combination, determining an application component of an application to be monitored, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of an IDE runtime

environment. In other words, neither Gove nor Karkare teach or suggest selecting an application component to be monitored where the component is selected from between operational logic and business logic. Furthermore, the Applicants respectfully contend that neither Karakre nor Gove teach or suggest a policy manager of an IDE user interface environment defining a policy specifying an operational concern to be monitored and a policy agent of an IDE runtime environment enforcing such a policy to determine metrics associated with a selected application component as recited in each independent claim. For example, see FIG. 1 and paragraph 32 of the pending application illustrating such recitations.

In an effort to demonstrate a teaching of selecting an application component to be monitored where the component is selected from between operational logic and business logic, the Office Action introduces Charisius. However, the Applicants respectfully contend that Charisius fails to correct the deficiencies of the other references. Charisius is directed to a software development tool that “allows a developer to simultaneously view a graphical and a textual display of source code” which “are synchronized so that a modification in one view is automatically reflected in the other view.” (paragraph 11) More specifically, Charisius describes:

“An improved software development tool which allows a developer to receive an identification of a data structure with an attribute field in a database, determine whether the data structure is associated with source code, if the data structure is associated with the source code, determine whether an attribute field of the data structure is associated with an attribute in the source code, and if the attribute field is not associated with the attribute in the source code, generate the attribute in the source code from the attribute field.” (Abstract)

However, the Applicants respectfully contend that Charisius fails to teach or suggest, alone or in combination with Karkare and/or Gove, selecting an application component to be monitored where the component is selected from between operational logic and business logic. Rather, Charisius describes “relating a data structure and object-oriented element”, i.e., attribute, beginning at paragraph 98 and more specifically at paragraphs 99-101 (cited by the

Office Action) notes that EJBs generated by a developer for the purpose of persisting a database table may include such information as customer and client information and may be invoked to provide such information to an application. However, the Applicants contend that this cannot reasonably be read as teaching or suggesting, alone or in combination with Karkare and/or Gove, selecting an application component to be monitored where the component is selected from between operational logic and business logic. Furthermore, Charisius does not teach or suggest, alone or in combination with Karkare and/or Gove, a policy manager of an IDE user interface environment defining a policy specifying an operational concern to be monitored and a policy agent of an IDE runtime environment enforcing such a policy to determine metrics associated with a selected application component as recited in each independent claim.

Claim 1, upon which claims 2-8 depend, recites in part “defining a policy specifying an operational concern to be monitored in the IDE runtime environment through a policy manager of the IDE user interface; determining an application component of an application to be monitored in the IDE based on the policy, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment; [and] monitoring the application component with a policy agent executing in the IDE runtime environment and enforcing the policy specifying the operational concern to determine a plurality of metrics associated with the application component.” None of the references teach or suggest, alone or in combination, determining an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment and monitoring the application component in the IDE runtime environment. Furthermore, none of the references teach or suggest, alone or in combination,

defining a policy specifying an operational concern to be monitored in the IDE runtime environment through a policy manager of the IDE user interface and monitoring the application component with a policy agent executing in the IDE runtime environment and enforcing the policy specifying the operational concern to determine a plurality of metrics associated with the application component. For at least these reasons, the Applicants respectfully request reconsideration and withdrawal of the rejection.

Claim 9 recites in part “define a policy specifying an operational concern to be monitored in an integrated development environment (IDE) runtime environment through a policy manager of an IDE user interface environment; determine an application component of an application to be monitored in the IDE runtime environment based on the policy, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment; [and] monitor the application component with a policy agent executing in the IDE runtime environment and enforcing the policy specifying the operational concern to determine a plurality of metrics associated with the application component.” None of the references teach or suggest, alone or in combination, determining an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment and monitoring the application component in the IDE runtime environment. Furthermore, none of the references teach or suggest, alone or in combination, defining a policy specifying an operational concern to be monitored in the IDE runtime environment through a policy manager of the IDE user interface and monitoring the application component with a policy agent executing in the IDE runtime environment and enforcing the policy specifying the operational concern to determine a plurality of metrics associated with the

application component. For at least these reasons, the Applicants respectfully request reconsideration and withdrawal of the rejection.

Claim 10, upon which claims 11-21 depend, recites in part “a component repository configured to maintain a list of available application components that can be invoked by an integrated development environment (“IDE”) runtime environment to monitor an application, the application comprising operational logic and business logic; an IDE runtime environment configured to open an application component, the IDE runtime environment comprising: a policy agent configured to monitor operation of the application component to determine a plurality of metrics associated with the application component by enforcing a policy specifying an operational concern to be monitored in the IDE, wherein the application component is selected from the operational logic of the application; and an IDE user interface environment configured to allow a user to perform software development tasks, the IDE user interface comprising: a policy manager configured to define the policy specifying the operational concern to be monitored in the IDE.” None of the references teach or suggest, alone or in combination, a component repository configured to maintain a list of available application components that can be invoked by an integrated development environment (“IDE”) runtime environment to monitor an application, the application comprising operational logic and business logic and monitoring operation of the application component to determine a plurality of metrics associated with the application component wherein the application component is selected from the operational logic of the application. Furthermore, none of the references teach or suggest, alone or in combination, the IDE runtime environment comprising: a policy agent configured to monitor operation of the application component to determine a plurality of metrics associated with the application component by enforcing a policy specifying an operational concern to be monitored in the IDE and the IDE user interface comprising a policy manager configured to define the policy specifying the operational concern to be monitored in the IDE. For at least these reasons, the Applicants respectfully request reconsideration and withdrawal of the rejection.

**CONCLUSION**

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance and an action to that end is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 303-571-4000.

Respectfully submitted,

/William J. Daley/  
William J. Daley  
Reg. No. 52,471

TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, Eighth Floor  
San Francisco, California 94111-3834  
Tel: 303-571-4000  
Fax: 415-576-0300

WJD:jep  
62409769 v1